So far we have considered 0-D & 1-D analytical transients. 0-D were "lumped capacitance" with very fast internal temperature response. 1-D is initially semi-infinite until the heat wave crosses the material extent at which point the problem requires a separation of variables solution & very simple requirements.

In many cases, analytical solutions, though accurate & fast, will be overwhelmingly complicated & infeasible. Where we have no choice but numerical solutions, Numerical Transient problems use a combination of our 1-D transient techniques:

1) Distribute nodes across the domain
2) Energy balance on each control volume will lead to a state equation for each node: \(
\frac{dT_i}{dt} = \sum (T_j - T_i)\)
3) Numerically integrate the state equations forward in time using one of several techniques (Euler's, Heun's, Crank-Nicolson, RK4, etc.)

State equation: provides the time rate of change of the state variable given its value & time (eg: temperature vs time).

We can't just construct a finite difference network through time. We have to consider trends at one point in time to predict to the next time.
Example: Most basic transient where a block of material is exposed to a step change in environmental temperature, $T_e$.

**Step 0:** Quick resistance calc $\Rightarrow$ $R_{\text{eq}} = \frac{\text{Resistance}}{\text{Loss due to}}$ $\frac{1}{\alpha}$

$B_i = \frac{h_i}{k}$ if $B_i < 0$, I then lump the system as both of material spatially.

Otherwise it's a 1D problem where we need to discretize the domain into small nodes and elements to move through time.

**Step 1:** Energy Balance $\Rightarrow$ $E_{\text{in}} = E_{\text{out}} + \frac{dE_{\text{stored}}}{dT} \Rightarrow 0 = \frac{dE_{\text{stored}}}{dT}$

**Step 2:** Rate Equations $\Rightarrow$ $\frac{d}{dt} (T - T_{\text{bs}}) \Rightarrow \frac{dT}{dt} = \frac{d}{dt} (T - T_{\text{bs}}) \Rightarrow \frac{dT}{dt} = \frac{-hA(T - T_{\text{bs}})}{PFc}$

$t_{\text{amp}}$ until this point our analytical & numerical approaches are usually the same.

**Step 3:** Discretize $\Rightarrow$ Define the domain by dividing the simulation time (domain) into true steps $\Rightarrow t_i = (i-1) \frac{t_{\text{sim}}}{M-1}$ for $i = 1 \ldots M$ & $\Delta t = \frac{t_{\text{sim}}}{M-1}$

**Step 4:** Apply a Numerical Integration Technique $\Rightarrow$ There are many to choose from, we'll start with the most basic & build in sophistication.
Explicit Techniques

Euler's Method: Computes the temperature at the end of each time step based on the temperature at the beginning of the step and the state equation. Euler's method assumes the rate of temperature change is constant within the time step and equal to the value at the beginning of the step.

\[ T_{i+1} = T_i + \frac{dT}{dt} \bigg|_{T_{i+1/2, t_{i+1}}} \Delta t \]

State equation assuming rate change at beginning is constant through time:

Because \( T_{i+1} \) can be calculated explicitly, using information at the beginning, Euler's method is known as an explicit technique.

\[ T_{i+1} = T_i - \frac{\Delta t}{\text{Cp}} (T_{i+1} - T_0) \]

Euler's Method, like all numerical techniques is not precise. If \( \Delta t \) is too large for the problem, the technique becomes unstable. Show Figure 3-3 pg 436.

Rearranging:

\[ T_{i+1} = T_i \left(1 - \frac{\Delta t}{\text{Cp}}\right) + T_0 \frac{\Delta t}{\text{Cp}} \]

this becomes unstable when this coefficient go negative.

Thus a critical time step can be defined where \( \Delta t \) is made so that \( \Delta t \ll \Delta t_{\text{crit}} \) for better accuracy.

Predictor-Corrector Techniques: Heun's Method & Runge-Kutta 4th Order (RK4)

Heun's method uses Euler's method to predict the temperature at the conclusion of a time step (\( \tilde{T}_{i+1} \)).

\[ T_{i+1} = T_i + \frac{\Delta t}{2} \left( \frac{dT}{dt} \bigg|_{T_{i+1/2, t_{i+1}}} + \frac{dT}{dt} \bigg|_{T_i} \right) \Delta t \]
The temperature at the end of the step $T_{n+1}$ then predicts the rate of temperature at the end of the step $\left( \frac{dT}{dT_{n+1}} \right)_{T_{n+1}}$. Then a corrector step averages the rates of change at the beginning and end of the step to predict $T_{n+1}$ actual.

$$T_{n+1} = T_n + \left[ \frac{dT}{dT_{n+1}} + \frac{dT}{dT_{n+1}} \right] \Delta t$$

$\Delta t$ is the average for each time step.

Implicit Techniques: Fully Implicit Crank-Nicolson and EES INTEGRAL functions require an implicit solver but avoid the critical timestep issue.

**Fully Implicit:**

$$T_{n+1} = T_n + \frac{dT}{dT_{n+1}} \Delta t \quad \Rightarrow \quad T_{n+1} = T_n - \left( \frac{dT}{dT_{n+1}} \right) \Delta t$$

Show Figure 3-7 for stability.

Crank-Nicolson: Combines Euler's method & Fully implicit method.

$$T_{n+1} = T_n + \left[ \frac{dT}{dT_{n+1}} + \frac{dT}{dT_{n+1}} \right] \Delta t \quad \Rightarrow \quad T_{n+1} = T_n - \left[ \frac{dT}{dT_{n+1}} + \frac{dT}{dT_{n+1}} \right] \Delta t$$

Combines high accuracy & stability.

EES INTEGRAL: Third Order semi-implicit with adaptive step-size complex.

Show accuracy plot of all techniques.